having memory including random access memory, the model being stored on a computer-readable media, the method comprising:

creating a scenegraph;

loading data of the first surface feature from a database stored in the data storage system into memory of the computer;

creating an aggregate feature for the first surface feature;

creating a first graphics object from the aggregate feature;

adding the first graphic object to the ~~application~~ scenegraph;

creating a geometry object for the aggregate feature;

editing the first surface feature in the model;

selectively updating the graphics for the model; and

removing the first graphics object of the first surface feature from the ~~application~~ scenegraph.

2. (Original) The method of claim 1, wherein said step of editing comprises:

providing an interface;

providing an IGM operative with said interface;

providing a GQI operative with the IGM;

through the interface, selecting an operation to perform on a second surface feature, the interface being constructed and arranged to inform the IGM of the selection;

invoking the operation with the GQI;

performing at least one callback from the GQI to the IGM during execution of the operation; and

updating the graphics object of the model to refresh the output device.

3. (Original) The method of claim 1, wherein the step of editing is accomplished via irregular space partitioning.

4. (Original) The method of claim 1, wherein the step of updating the graphics object includes creating the graphics object.

5. (Original) The method of claim 2, wherein the step of performing the callback causes a change of state for the aggregate feature.

6. (Original) The method of claim 5, wherein the change of state for the aggregate feature is in a consistency finite state machine.

7. (Original) The method of claim 6, wherein the consistency finite state machine manages consistency between geometry and graphics.

8. (Original) The method of claim 5, wherein the step of performing the callback includes invalidating the graphics object for the aggregate feature.

9. (Original) The method of claim 5, wherein the step of performing the callback includes validating the geometry of the aggregate feature.

10. (Original) The method of claim 5, wherein the step of performing the callback includes providing a set of objects comprising:

a first geometry modeler feature object;

a changed geometry object contained in the geometry modeler feature object;

a first meta-property attribute object operatively associated with the changed geometry object;

a first meta-property object associated with the first meta-property attribute object, the first meta-property object having:

a point set preservation property object;

a point set preservation property policy object;

a cell back pointer property object;

an aggregate back pointer property policy object;

a geometry cell object associated with the cell back pointer object; an aggregate cell object associated with the geometry cell object; and

a display cell graphics object associated with the aggregate cell object;

a second meta-property attribute object associated with the geometry modeler feature object; and

a second meta-property object associated with the second meta-property attribute object, the second meta-property object having:

a second point set preservation property object;

a second point set preservation property policy object;

a feature back pointer property object;

a second aggregate back pointer property policy object;

a geometry feature object associated with the feature back pointer property object;

an aggregate feature object associated with the geometry feature object; and

a display feature graphics object associated with the aggregate feature object.

11. (Original) The method of claim 10, wherein the step of performing the callback includes:

if the callback is for a volume object, then performing an edit callback for a volume feature; otherwise performing an edit callback for a surface feature.

12. (Original) The method of claim 1, wherein the step of updating includes updating graphics for only those cells that have changed geometrically.

13. (Original) The method of claim 1, wherein the step of updating includes updating graphics for only those features that have changed topologically.

14. (Original) The method of claim 1, wherein the step of creating graphics for at least one aggregate feature from the model comprises:

obtaining all relevant surface features in the model;

for each of the relevant surface features:

if the surface feature has a graphics object, then updating the graphics object for the surface feature, otherwise creating a graphics object for the surface feature; and

adding the graphics object for the surface feature to a surface scenegraph root node.

15. (Original) The method of claim 14, wherein the step of creating the graphics object comprises:

obtaining aggregate objects of all two-dimensional cells for the surface feature;

prompting the aggregate objects of each of the cells to obtain at least one valid graphics object and to add the graphics object to a sub-scenegraph of the surface feature; and

validating the graphics object of the surface feature.

16. (Original) The method of claim 15, wherein the step of obtaining valid graphics object for a cell comprises:

if the graphics object for the cell does not exist, then creating the graphics object for the cell and validating the graphics; and

if the graphics object for the cell exists and is not valid, then updating the graphics object for the cell and validating the graphics.

17. (Original) The method of claim 1, wherein the step of updating the graphics object includes referencing a visibility finite state machine.

18. (Original) The method of claim 17, wherein the visibility finite state machine manages the updating of the graphics objects and the updating of the visibility of the graphics objects.

19. (Original) The method of claim 18, wherein the visibility finite state machine enables the update of only those graphics of objects that are designated as visible.

20. (Original) The method of claim 18, wherein the step of updating includes:

designating a graphics object as visible;

checking the validity of the graphics object;

if the graphics object is valid, then adding the graphics object to the scenegraph, otherwise, updating the graphics object and then adding the graphics object to the scenegraph.

21. (Original) The method of claim 1, wherein the step of creating graphics for at least one volume cell from the model comprises:

obtaining at least one relevant active volume cell in the model;

ensuring that graphics objects of all two-dimensional cells of each volume cell have been created;

for each of the two-dimensional cells, obtaining the aggregates and create a new aggregate if the two-dimensional cell aggregate does not exist;

ensuring that each two-dimensional cell has valid graphics;

creating a graphics object for each of the at least one volume cell;

for each volume cell, adding the graphics content of each two-dimensional cells of the volume cell to the sub-scenegraph of the graphics object of the volume cell; and

adding the graphics object for each volume cell to the volume scenegraph root node.

22. (Original) The method of claim 21, wherein an instantiation of a graphics content, of a two-dimensional cell is shared by at least one scenegraph containing a graphics object of a surface feature that has the two-dimensional cell as a child, and at least one of the scenegraphs that contains graphics objects of either one volume cell or two volume cells that have the two-dimensional cell as part of their boundaries.

23. (Original) The method of claim 6, wherein the step of updating includes:

checking the state of each feature in the scenegraph;

if the graphics is valid or if the geometry is invalid for the feature, then not updating the graphics object of the feature;

if the geometry is valid and the graphics are invalid, then updating the graphics object of the feature.

24. (Original) The method of claim 6, wherein the step of updating includes:

checking the state of each cell in the scenegraph;

if the graphics is valid or if the geometry is invalid for the cell, then not updating the graphics object of the cell;

if the geometry is valid and the graphics is invalid, then updating the graphics object of the cell.

25. (Original) The method of claim 11, wherein the step of performing an edit callback for a volume feature comprises:

registering the meta-property split callback class method with the geometry modeler interface, the geometry modeler interface constructed and arranged to be invoked when a volume split event occurs;

attaching a first meta-property attribute to at least one volume object contained by the volume feature;

receiving, from the geometry modeler interface, a callback specifying a first volume object, a second volume affected by a change to the first volume object and the first meta-property attribute;

obtaining a pointer value from the first meta-property attribute and de-referencing the pointer value to locate a first meta-property object;

invoking a split callback in the first meta-property object with the first meta-property attribute, the first volume object and the second volume object, comprising:

obtaining a first point set preservation property instance;

obtaining a first point set preservation policy instance from the property instance;

initiating a first split callback to the point set preservation policy instance with the point set preservation property instance, the first volume object and the second volume object comprising:

obtaining at least one containing feature for the first volume object; and

initiating a feature add child update on the geometry modeler interface with the containing feature and the second volume object;

obtaining a cell back pointer property instance; and

obtaining an aggregate back pointer property policy instance from the cell back pointer property instance; and initiating a second split callback to the aggregate back pointer property policy instance with the cell back pointer property instance comprising:

obtaining a volume geometry cell object from the cell back pointer property instance; and initiating a cell split call to the volume geometry cell object, comprising:

initiating a call to the volume cell aggregate patron of the volume geometry cell object to invalidate the graphics of the first volume.

26. (Original) The method of claim 25, wherein the step of performing a feature add child callback for a volume feature comprises:

registering the meta-property add child callback class method with the geometry modeler interface to be invoked when a feature add child event occurs;

attaching a second meta-property attribute instance to the volume feature;

receiving from the geometry modeler interface the add child callback specifying the volume feature, a volume object and the second meta-property attribute;

obtaining a pointer value from the second meta-property attribute and de-referencing the pointer value to locate a second meta-property object;

invoking the add child callback in the second meta-property object comprising:

obtaining a second point set preservation property instance;

obtaining a second point set preservation property policy instance from the second point set preservation property instance; and

initiating the add child callback method of the second point set policy object with the volume feature and the volume geometry object, comprising:

attaching the point set preservation property to the volume cell;

obtaining a feature back pointer property instance;

obtaining a second aggregate back pointer property policy instance from the feature back pointer property instance; and initiating the add child callback method of the second aggregate back pointer property policy instance with the volume feature, the volume geometry object and the volume feature back pointer property, comprising:

initiating an add child notify call to the volume feature geometry object identified by the feature back pointer property instance, comprising:

initiating a call to the volume feature aggregate patron of the volume feature object to validate the geometry of the volume feature object; and

initiating a call to the volume feature aggregate patron of the volume feature object to invalidate the graphics of the volume feature object.

27. (Original) The method of claim 11, wherein the step of performing an edit callback for a surface feature comprises:

registering the meta-property split callback class method with the geometry modeler interface to be invoked when a surface split event occurs;

attaching a first meta-property attribute to at least one surface object contained by the surface feature;

receiving from the geometry modeler interface a callback specifying a first surface object, a second surface affected by a change to the first surface and the first meta-property attribute;

obtaining a pointer value from the first meta-property attribute and de-referencing the pointer value to locate a first meta-property object;

invoking a split callback in the first meta-property object with the first surface object, the second surface object and the first meta-property attribute, comprising:

obtaining a first point set preservation property instance;

obtaining a first point set preservation policy instance from the property instance; and

initiating a first split callback to the point set preservation policy instance with the point set preservation property instance, the first surface object and the second surface object comprising:

obtaining at least one containing feature for the first surface object; and

initiating a feature add child update on the geometry modeler interface with the containing feature and the second surface object;

obtaining a cell back pointer property instance;

obtaining an aggregate back pointer property policy instance from the cell back pointer property instance; and

initiating a second split callback to the aggregate back pointer property policy instance with the cell back pointer property instance comprising:

obtaining a surface geometry cell object from the cell back pointer property instance; and

initiating a cell split call to the surface geometry cell object, comprising:

initiating a call to the surface cell aggregate patron of the surface geometry cell object to invalidate the graphics of the first surface.

28. (Original) The method of claim 27, wherein the step of performing a feature add callback comprises:

registering the meta-property add child callback class method with the geometry modeler interface to be invoked when a feature add child event occurs;

attaching a second meta-property attribute instance to the surface feature;

receiving from the geometry modeler interface the add child callback specifying the surface feature, a surface object and the second meta-property attribute;

obtaining a pointer value from the second meta-property attribute and de-referencing the pointer value to locate a second meta-property object; and

invoking the add child callback in the second meta-property object comprising:

obtaining a second point set preservation property instance;

obtaining a second point set preservation property policy instance from the second point set preservation property instance; and

initiating the add child callback method of the second point set policy object with the surface feature and the surface geometry object, comprising:

attaching the point set preservation property to the surface cell;

obtaining a feature back pointer property instance;

obtaining a second aggregate back pointer property policy instance from the feature back pointer property instance; and

initiating the add child callback method of the second aggregate back pointer property policy instance with the surface feature, the surface geometry object and the surface feature back pointer property, comprising:

initiating an add child notify call to the surface feature geometry object identified by the feature back pointer property instance, comprising:

initiating a call to the surface feature aggregate patron of the surface feature object to validate the geometry of the surface feature object; and

initiating a call to the surface feature aggregate patron of the surface feature object to invalidate the graphics of the surface feature object.

29. (Original) The method of claim 2, wherein the step of performing the callback causes a change of state for the cell.

30. (Original) The method of claim 29, wherein the change of state for the cell is in a consistency finite state machine.

31. (Original) The method of claim 2, wherein the second surface feature is not contained within the model.

32. (Original) The method of claim 2, wherein the second surface feature is contained within the model.

33. (Original) The method of claim 1, wherein the output device is a display.

34. (Original) The method of claim 11, wherein the step of performing an edit callback for a volume feature comprises:

registering the meta-property merge callback class method with the geometry modeler interface to be invoked when a volume merge event occurs;

attaching a first meta-property attribute to at least one volume object contained by the volume feature;

receiving from the geometry modeler interface a callback specifying a first volume object, a second volume object, a surface object which formerly bounded the first and second volume objects and which has been removed from the model, and a meta-property attribute;

obtaining a pointer value from the geometry model attribute and de-referencing the pointer value to locate a first meta-property object; and

invoking a merge callback in the first meta-property object, comprising:

obtaining a first point set preservation property instance;

obtaining a first point set preservation policy instance from the property instance; and

initiating a first merge callback to the point set preservation policy instance with the point set preservation property instance, the first volume object, the second volume object, and the surface object, comprising:

obtaining at least one containing feature for the first volume object; and

initiating a feature remove child update on the geometry modeler interface with the containing feature and the second volume object;

obtaining a cell back pointer property instance;

obtaining an aggregate back pointer property policy instance from the cell back pointer property instance; and

initiating a second merge callback to the aggregate back pointer property policy instance with the cell back pointer property instance, the first volume object, the second volume object, and the surface object, comprising:

obtaining a volume geometry cell object from the cell back pointer property instance, and initiating a cell merge call to the volume geometry cell object, comprising:

initiating a call to the volume cell aggregate patron of the volume geometry cell object to invalidate the graphics of the first volume.

35. (Original) The method of claim 34, wherein the step of performing a feature remove child callback comprises:

registering the meta-property remove child callback class method with the geometry modeler interface to be invoked when a feature remove child event occurs;

attaching a second meta-property attribute instance to the volume feature;

receiving from the geometry modeler interface the remove child callback specifying the volume feature, a volume object and the second meta-property attribute;

obtaining a pointer value from the second meta-property attribute and de-referencing the pointer value to locate a second meta-property object; and

invoking the remove child callback in the second meta-property object comprising:

obtaining a second point set preservation property instance;

obtaining a second point set preservation property policy instance from the second point set preservation property instance;

initiating the remove child callback method of the second point set policy object with the volume feature and the volume geometry object, comprising:

removing the point set preservation property from the volume cell;

obtaining a feature back pointer property instance;

obtaining a second aggregate back pointer property policy instance from the feature back pointer property instance; and initiating the remove child callback method of the second aggregate back pointer property policy instance with the volume feature, the volume geometry object and the volume feature back pointer property, comprising:

initiating an remove child notify call to the volume feature geometry object identified by the feature back pointer property instance, comprising:

initiating a call to the volume feature aggregate patron of the volume feature object to validate the geometry of the volume feature object; and

initiating a call to the volume feature aggregate patron of the volume feature object to invalidate the graphics of the volume feature object.

36. (Original) The method of claim 11, wherein the step of performing an edit callback for a surface feature comprises:

registering the meta-property merge callback class method with the geometry modeler interface to be invoked when a surface merge event occurs;

attaching a first meta-property attribute to at least one surface object contained by the surface feature;

receiving from the geometry modeler interface a callback specifying a first surface object, a second surface object, and a curve object which formerly bounded the first and second surfaces and which has been removed from the model and a meta-property attribute;

obtaining a pointer value from the geometry model attribute and de-referencing the pointer value to locate a first meta-property object; and invoking a merge callback in the first meta-property object, comprising:

obtaining a first point set preservation property instance;

obtaining a first point set preservation policy instance from the property instance; and

initiating a first merge callback to the point set preservation policy instance with the point set preservation property instance, the first surface object, the second surface object, and the curve object, comprising:

obtaining at least one containing feature for the first surface object; and

initiating a feature remove child update on the geometry modeler interface with the containing feature and the second surface object;

obtaining a cell back pointer property instance;

obtaining an aggregate back pointer property policy instance from the cell back pointer property instance; and

initiating a second merge callback to the aggregate back pointer property policy instance with the cell back pointer property instance, the first surface object, the second surface object, and the curve object comprising:

obtaining a surface geometry cell object from the cell back pointer property instance; and

initiating a cell merge call to the surface geometry cell object, comprising:

initiating a call to the surface cell aggregate patron of the surface geometry cell object to invalidate the graphics of the first surface.

37. (Original) The method of claim 36, wherein the step of performing a feature remove callback comprises:

registering the meta-property remove child callback class method with the geometry modeler interface to be invoked when a feature remove child event occurs;

attaching a second meta-property attribute instance to the surface feature;

receiving from the geometry modeler interface the remove child callback specifying the surface feature, a surface object and the second meta-property attribute;

obtaining a pointer value from the second meta-property attribute and de-referencing the pointer value to locate a second meta-property object; and

invoking the remove child callback in the second meta-property object with the second meta-property attribute, the surface feature and the surface geometry object comprising:

obtaining a second point set preservation property instance;

obtaining a second point set preservation property policy instance from the second point set preservation property instance; and

initiating the remove child callback method of the second point set policy object with the surface feature and the surface geometry object, comprising:

removing the point set preservation property from the surface cell;

obtaining a feature back pointer property instance;

obtaining a second aggregate back pointer property policy instance from the feature back pointer property instance; and

initiating the remove child callback method of the second aggregate back pointer property policy instance with the surface feature, the surface geometry object and the surface feature back pointer property, comprising:

initiating an remove child notify call to the surface feature geometry object identified by the feature back pointer property instance, comprising:

initiating a call to the surface feature aggregate patron of the surface feature object to validate the geometry of the surface feature object; and

initiating a call to the surface feature aggregate patron of the surface feature object to invalidate the graphics of the surface feature object.

38. (Currently amended) A computer system for interactively editing a model stored on computer-readable media and having a first surface, the computer system further having a processor, a data storage system, at least one input device, and at least one output device, the computer system further having random access memory constructed and arranged to contain an object structure, the object structure comprising:

a geometry query interface object, the geometry query interface object having a GQI material property framework object, the GQI material property framework object comprising:

a cc_RefObj object;

a gmMP object having an IsA relationship with the cc_RefObj object;

a gmMPPolicy object having an IsA relationship with the cc_RefObj object; and

a gmMPConstant object having an IsA relationship with the gmMP object

an interactive geometric modeling object derived from a common model builder object, the interactive geometric modeling object having a relationship with the GQI material property framework object, and an IGM material property framework object, the IGM material property framework object comprising:

a gmMPPolyXYZ object having an IsA relationship with the gmMP object;

a gmMPZ object having an IsA relationship with the gmMP object;

a gmMPTime object having an IsA relationship with the gmMPZ object;

a gmMPDepth object having an IsA relationship with the gmMPZ object;

a gmMP2DPoly object having an IsA relationship with the gmMP object;

a gmMP2DGrid object having an IsA relationship with the gmMP object;

a gmMP3dGrid object having an IsA relationship with the gmMP object;

a gmMPName object having an IsA relationship with the gmMPConstant object;

an mbCellGMReference object having an IsA relationship with the gmMPConstant object;

a mbFtrGMReference object having an IsA relationship with the gmMPConstant object;

a vspQualityProp object having an IsA relationship with the gmMPConstant object;

a vspTransverseIsotroppy object having an IsA relationship with the gmMPConstant object; and

a fbFtrParameters object having an IsA relationship with the gmMPConstant object;

wherein data is processed by the object structure in order to enable a user to edit the model ~~stored in the database~~.

39. (Original) The computer system as in claim 38, the IGM material property framework object further comprising:

a gmGradientProp object;

a vspDensProp object having an IsA relationship with the gmGradProp object;

a vspVelPProp object having an IsA relationship with the gmGradProp object;

a vspVelSProp object having an IsA relationship with the gmGradProp object; and

a gmResistivityProp object having an IsA relationship with the gmGradProp object.

40. (Original) The computer system as in claim 38, the IGM material property framework object further comprising:

a gmMPNamePropertyPolicy object having a composition relationship with the gmMPName object.

41. (Original) The computer system as in claim 38, the IGM material property framework object further comprising:

a gmMPIGMPropertyPolicy object having a composition association with the mbCellGMReference object and a composition relationship with the mbFtrGMReference object.

42. (Original) The computer system as in claim 38, the GQI material property framework object further comprising:

a gmutil_status object;

an mv_vt object; and

a gm_mp_atoms object.

43. (Original) The computer system as in claim 38, the GQI material property framework object further comprising:

a gmSysRules object having an IsA relationship with the gmMPConstant object;

a gmSysPSPProperty object having an IsA relationship with the gmMPConstant object;

a gmUtilFtrBnd object having an IsA relationship with the gmMPConstant object;

a gmMPGQIPolicy object having an IsA relationship with the gmMPPolicy object;

a gmSysPSPolicy object having an IsA relationship with the gmMPPolicy object and a composition relationship with the gmSysPSPProperty object;

a gmSysRulesPolicy object having an IsA relationship with the gmMPPolicy object and a composition relationship with the gmSysRules object; and

a gmMPVolumePropertyPolicy object having an IsA relationship with the gmMPPolicy object and a composition relationship with the gmMPConstant object.

44. (Original) The computer system as in claim 43, the GQI material property framework object further comprising:

a gmUtilFtrBndPolicy object having an IsA relationship with the gmMPGQIPolicy object.

45. (Original) The computer system as in claim 43, the GQI material property framework object further comprising:

a gmMPTopologyTraversalStates object having a composition relationship with the gmMPPolicy object.

46. (Original) The computer system as in claim 44, wherein a gmMPIGMPropertyPolicy object has an IsA relationship with the gmUtilFtrBndPolicy object.

47. (Original) The computer system as in claim 43, wherein a gmMPNamePropertyPolicy object has an IsA relationship with the gmMPVolumePropertyPolicy object.

48. (Original) The computer system as in claim 39, wherein a gmMPVolumePropertyPolicy object has a first composition relationship with the gmGradientProp object and a second composition relationship with the gmMPZ object.

49. (Original) The computer system of claim 38, the computer system further comprising:

an aqi_Parameter object having a relationship with the gmMP object;

a gmMPTopologyTraversalState object having a relationship with the gmMPPolicy object;

a first gqi_MetaProperty object having a relationship with the gmMP object;

a second gqi_MetaProperty object having a relationship with the gmMP object;

a gqi_AttachmentSite object having a relationship with the first gqi_MetaProperty object; and

a gqi_Core object having a relationship with the second gqi_MetaProperty object and an IsA relationship with the gqi_AttachmentSite object.

50. (Original) The computer system of claim 49, the computer system further comprising:

an ag_BaseClass object;

an ag_Geometry object having an IsA relationship with the ag_BaseClass object;

a gm_Geometry object having an hasA relationship with the ag_Geometry object, the gm_Geometry object having a relationship with the Geometry object;

an oi_Feature object having a relationship with an ag_Feature object, the ag_Feature object having an IsA relationship with the ag_Geometry object, the ag_Feature object further having a relationship with the gm_Feature object;

an oi_Cell object having a relationship with an ag_Cell object, the ag_Cell object having an IsA relationship with the ag_Cell object, the ag_Cell object further having a relationship with the gm_Cell object;

an X_Feature object having a relationship with the gm_Feature object, the X_Feature object having an IsA relationship with an xAttrFtr object, the xAttrFtr object having an IsA relationship with a gqi_Feature object, and the gqi_Feature object having an IsA relationship with the X_Feature object;

an X_Cell object, the X_Cell object having an IsA relationship with the X_Feature object, the X_Cell object having an IsA relationship with a first xAttrMP object, the first xAttrMP object having an IsA relationship with a first gqi_MetaProperty object, the first gqi_MetaProperty object having an association with a FtrRef object, the FtrRef object having at least one association with the gm_Feature object; and

the X_Feature object further having an IsA relationship with a second xAttrMP object, the second xAttrMP object having an IsA relationship with a second gqi_MetaProperty object, the second gqi_MetaProperty object having an association with CellRef object, the CellRef object having at least one association with the gm_Cell object.

51. (Currently amended) A random access memory, the random access memory having an object structure comprising:

a geometry query interface object, the geometry query interface object having a GQI material property framework object, the GQI material property framework object comprising:

a cc_RefObj object;

a gmMP object having an IsA relationship with the cc_RefObj object;

a gmMPPolicy object having an IsA relationship with the cc_RefObj object; and

a gmMPConstant object having an IsA relationship with the gmMP object

an interactive geometric modeling object derived from a common model builder object, the interactive geometric modeling object having a relationship with the GQI material property framework object, and an IGM material property framework object, the IGM material property framework object comprising:

a gmMPPolyXYZ object having an IsA relationship with the gmMP object;

a gmMPZ object having an IsA relationship with the gmMP object;

a gmMPTime object having an IsA relationship with the gmMPZ object;

a gmMPDepth object having an IsA relationship with the gmMPZ object;

a gmMP2DPoly object having an IsA relationship with the gmMP object;

a gmMP2DGrid object having an IsA relationship with the gmMP object;

a gmMP3dGrid object having an IsA relationship with the gmMP object;

a gmMPName object having an IsA relationship with the gmMPConstant object;

an mbCellGMReference object having an IsA relationship with the gmMPConstant object;

a mbFtrGMReference object having an IsA relationship with the gmMPConstant object;

a vspQualityProp object having an IsA relationship with the gmMPConstant object;

a vspTransverseIsotroppy object having an IsA relationship with the gmMPConstant object; and

a fbFtrParameters object having an IsA relationship with the gmMPConstant object;

wherein data is processed by the object structure in order to enable a user to edit ~~the~~ a model stored in ~~the~~ a database.

52. (Original) The random access memory as in claim 51, the IGM material property framework object further comprising:

a gmGradientProp object;

a vspDensProp object having an IsA relationship with the gmGradProp object;

a vspVelPProp object having an IsA relationship with the gmGradProp object;

a vspVelSProp object having an IsA relationship with the gmGradProp object; and

a gmResistivityProp object having an IsA relationship with the gmGradProp object.

53. (Original) The random access memory as in claim 51, the IGM material property framework object further comprising:

a gmMPNamePropertyPolicy object having a composition relationship with the gmMPName object.

54. (Original) The random access memory as in claim 51, the IGM material property framework object further comprising:

a gmMPIGMPropertyPolicy object having a composition association with the mbCellGMReference object and a composition relationship with the mbFtrGMReference object.

55. (Original) The random access memory as in claim 51, the GQI material property framework object further comprising:

a gmutil_status object;

an mv_vt object; and

a gm_mp_atoms object.

56. (Original) The random access memory as in claim 51, the GQI material property framework object further comprising:

a gmSysRules object having an IsA relationship with the gmMPConstant object;

a gmSysPSPProperty object having an IsA relationship with the gmMPConstant object;

a gmUtilFtrBnd object having an IsA relationship with the gmMPConstant object;

a gmMPGQIPolicy object having an IsA relationship with the gmMPPolicy object;

a gmSysPSPolicy object having an IsA relationship with the gmMPPolicy object and a composition relationship with the gmSysPSPProperty object;

a gmSysRulesPolicy object having an IsA relationship with the gmMPPolicy object and a composition relationship with the gmSysRules object; and

a gmMPVolumePropertyPolicy object having an IsA relationship with the gmMPPolicy object and a composition relationship with the gmMPConstant object.

57. (Original) The random access memory as in claim 56, the GQI material property framework object further comprising:

a gmUtilFtrBndPolicy object having an IsA relationship with the gmMPGQIPolicy object.

58. (Original) The random access memory as in claim 56, the GQI material property framework object further comprising:

a gmMPTopologyTraversalStates object having a composition relationship with the gmMPPolicy object.

59. (Original) The random access memory as in claim 57, wherein a gmMPIGMPropertyPolicy object has an IsA relationship with the gmUtilFtrBndPolicy object.

60. (Original) The random access memory as in claim 56, wherein a gmMPNamePropertyPolicy object has an IsA relationship with the gmMPVolumePropertyPolicy object.

61. (Original) The random access memory as in claim 52, wherein a gmMPVolumePropertyPolicy object has a first composition relationship with the gmGradientProp object and a second composition relationship with the gmMPZ object.

62. (Original) The random access memory of claim 51, the random access memory further comprising:

an aqi_Parameter object having a relationship with the gmMP object;

a gmMPTopologyTraversalState object having a relationship with the gmMPPolicy object;

a first gqi_MetaProperty object having a relationship with the gmMP object;

a second gqi_MetaProperty object having a relationship with the gmMP object;

a gqi_AttachmentSite object having a relationship with the first gqi_MetaProperty object; and

a gqi_Core object having a relationship with the second gqi_MetaProperty object and an IsA relationship with the gqi_AttachmentSite object.

63. (Original) The random access memory of claim 62, the random access memory further comprising:

an ag_BaseClass object;

an ag_Geometry object having an IsA relationship with the ag_BaseClass object;

a gm_Geometry object having an hasA relationship with the ag_Geometry object, the gm_Geometry object having a relationship with the Geometry object;

an oi_Feature object having a relationship with an ag_Feature object, the ag_Feature object having an IsA relationship with the ag_Geometry object, the ag_Feature object further having a relationship with the gm_Feature object;

an oi_Cell object having a relationship with an ag_Cell object, the ag_Cell object having an IsA relationship with the ag_Cell object, the ag_Cell object further having a relationship with the gm_Cell object;

an X_Feature object having a relationship with the gm_Feature object, the X_Feature object having an IsA relationship with an xAttrFtr object, the xAttrFtr object having an IsA relationship with a gqi_Feature object, and the gqi_Feature object having an IsA relationship with the X_Feature object;

an X_Cell object, the X_Cell object having an IsA relationship with the X_Feature object, the X_Cell object having an IsA relationship with a first xAttrMP object, the first xAttrMP object having an IsA relationship with a first gqi_MetaProperty object, the first gqi_MetaProperty object having an association with a FtrRef object, the FtrRef object having at least one association with the gm_Feature object; and

the X_Feature object further having an IsA relationship with a second xAttrMP object, the second xAttrMP object having an IsA relationship with a second gqi_MetaProperty object, the second gqi_MetaProperty object having an association with CellRef object, the CellRef object having at least one association with the gm_Cell object.

64. (New) A method as in claim 1 wherein the in the loading step, the first surface feature from a database stored in the data storage system into the random access memory of the computer.

65. (New) A method as in claim 1 wherein the in the loading step, the first surface feature from a database stored in the data storage system into memory of the computer other than the random access memory.